RESEARCH ARTICLE

RSP Science Hub

International Research Journal on Advanced Science Hub
2582-4376
www.rspsciencehub.com
Vol. 07, Issue 11 November

Check for updates

http://dx.doi.org/10.47392/IRJASH.2025.107

Real-Time Data Engineering for Smart Applications

Souvari Ranjan Biswal[1]

[1]*Symbiosis International University, Pune, India.*

Article history

Received: 13 September 2025
Accepted: 06 October 2025
Published:07 November 2025

Keywords:

Real-Time Data Engineering; Smart Applications; Stream Processing; Edge Computing; Data Pipeline Architecture

Abstract

The digital transformation era is characterized by intelligent applications that increasingly rely on rapid processing and analysis of large, non-homogeneous data streams. These smart systems are built on real-time data engineering, enabling seamless data ingestion, transformation, and decision-making across industries such as healthcare, transportation, manufacturing, and finance. The paper will look into the architectural principles, enabling technologies, practical uses, and the future trend of real-time data engineering. It underscores that frameworks of stream processing, edge computing, and cloud-native processing and AI integration are critical to the development of scalable and responsive intelligent systems. The paper also lists popular traps like latency, scalability, data quality, and security, and also provides a projection of the future trend, which has been brought about by federated learning, data meshes, and quantum computing. The provided insights make a detailed picture of the development of the operational and strategic landscape of smart technologies in real-time by applying data engineering.

1. Introduction

The increasing popularity of digital technologies in the everyday life of people has led to the appearance of smart applications, the features of which are the presence of opportunities to process significant streams of data in a real-time manner, adapt to the behavior of users, and make independent decisions. Smart health monitoring devices, industrial automation, and intelligent transportation systems are just a few of the applications that require a robust, scalable, and efficient data infrastructure that will offer real-time data processing, decision-making, and responsiveness. These intelligent applications are therefore based on real-time data engineering. It enables the possibility to assemble, transform, and transfer information in the quickest possible as well as quicker than actual-world actions occur [1][2]. The pressure to offer real-time data streams due to the physical structure of the city being replaced by the physical infrastructures of smart systems, healthcare, finance, manufacturing, and other industries increases. The intersection of edge computing, technologies of big data, and artificial intelligence supports this change since, when combined, smart applications cannot only perceive everything around them but also intelligently act in it. This is made possible through real-time data engineering that monitors the whole process of information flow, wherein data are collected, pre-processed, analyzed, and transmitted to the consumers within milliseconds [3][4]. In this case, it should be noted that the evolving form and functionality of real-time data engineering is to plan and develop intelligent applications. The paper explains the key components, questions, and prospects of real-time data engineering and provides one of the concepts of how they are used to make smart systems more stable and responsive. I will discuss the architectural blueprints of real-time data engineering, technology that enables it to be applied, challenges and limitations of these systems, and future trends that will be defining the

field in the coming paragraphs. Passing on to this preliminary understanding of the concept, we move on to examine the design behind real-time data engineering systems and the major components and data flow processes that are required to enable real-time operations.

2. Real-Time Data Engineering Architectural Foundations

The architecture of a real-time data engineering system is designed in such a manner to ensure timely and reliable raw data to actionable intelligence, as shown in Figure 1. These architectures are decomposed on the foundations of the data ingestion layers, stream computing engines, storage engines, and output interfaces, which create a pipeline that employs the high velocity, high volume data with low latency [5][6]. Consumption of information is normally processed by a distributed system that can access it from a heterogeneous source of information that can include IoT sensors, mobile applications, social media services, and enterprise software. Such feeds may produce structured, semi-structured, or unstructured data, and, therefore, to enable a seamless integration, an elastic ingestion platform, such as Apache Kafka, Flume, or Amazon Kinesis, is required. The ingestion layer is often deployed in a manner that is scalable and fails gracefully since the existence of a bottleneck at this stage can compromise the responsiveness of downstream processing units [7]. Once data has been ingested, it is then forwarded to the real-time processing layer, where the stream processing frameworks, e.g., Apache Flink, Spark streaming, and Storm, are employed to transform, aggregate, and filter streams of data. The reason why these tools are chosen is because of the low-latency processing option and the capability to support complex event processing (CEP) that enables the system to identify patterns, correlation, and anomalies in the data flow [8][9]. The output of this layer is then passed directly to the downstream applications, or stored temporarily in the distributed and in-memory data stores such as Redis or Apache Druid, where they can be fast queried and run with analytics. To ensure that insights created through data are credible, a layer of real-time analytics, which does not tie to machine learning models, ought to be included in the modern architecture. Such models can be trained in batches and deployed to make a real-time inference to enable applications to be capable of making predictions and decisions as the data is being processed.

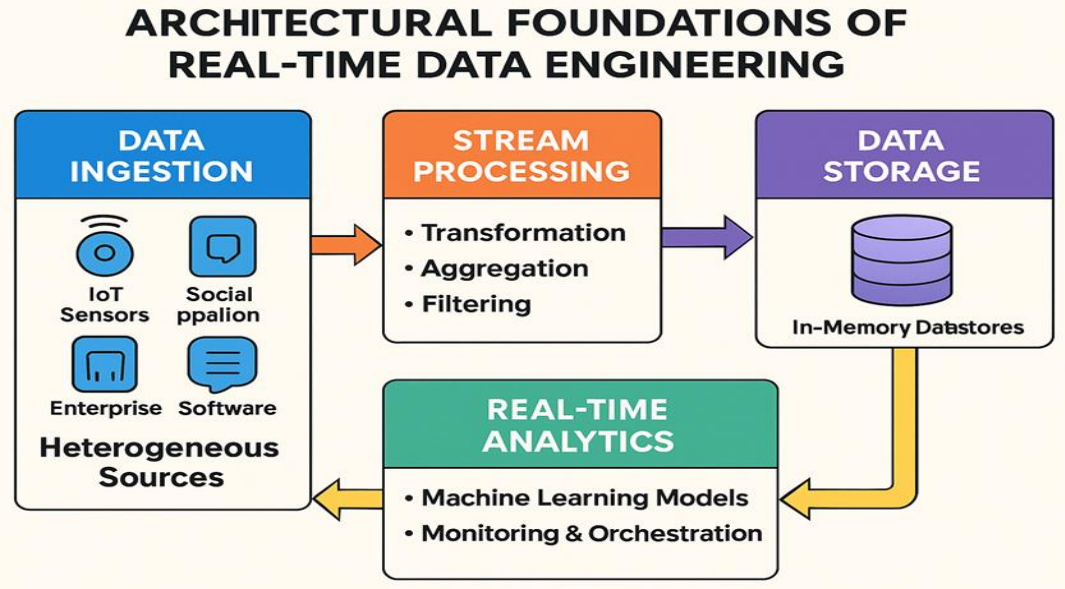

Figure 1 Architectural Foundations of Real-Time Data Engineering

This requires the functionality to feed ML models into the data pipelines; this encompasses model lifecycle management, model versioning, and real-time inference engines such as Tensorflow Serving or ONNX runtime [10]. Along with it, the instruments of orchestration, such as Kubernetes and Apache Airflow, are deployed to manage and scale containerized microservices in a manner that the various components of the data pipeline would cooperate and be resilient. There are also other

tools that could be employed to operate a system and monitor its health, provide warnings and visualization to ensure that the essentials of a team maintain the performance of a system at reasonable levels, considering the varying workloads [11]. It is in this architectural backdrop that we shall address the enabling technologies that support the real-time data engineering life cycle and make these complex pipelines a reality in real-life smart applications.

3. Enabling Technologies for Real-Time Smart Systems

The use of real-time data pipelines in smart applications will be highly reliant on the integration of the latest technologies that enable the ingestion, processing, analytics, and delivery of data. Not only do such technologies impact the performance and accuracy of smart systems, but they also determine the scalability, reliability, and efficiency of smart systems in dynamic environments. The concept of edge computing is also relevant in the removal of latency because the information is computed closer to the source, thereby doing away with the delays in communicating huge amounts of information to the central cloud servers. The first level of data processing can be used to filter, aggregate, and process data, which is then transmitted only to central nodes in applications such as autonomous driving or smart grid monitoring, where milliseconds can be paramount [12] [13]. Cloud-native platforms are other enabled systems that complement edge computing by providing on-demand and elastic compute and storage. Serverless cloud stream processing systems like AWS Lambda, Azure Stream Analytics, and Google Cloud Dataflow enable developers to automatically implement serverless stream processing functions, which automatically scale with workload. This paradigm also facilitates the maintenance of infrastructure because data engineers can focus on logic and not on the provisioning [14]. The other disruptive trend is the integration of artificial intelligence (AI) in real-time pipelines, where intelligent applications can be used to create insights, predict results, and enable the decision-making process to be automated. The data flow is increasingly being modified dynamically in response to new data by frameworks like Apache Beam or Kafka streams to incorporate AI models into the data flow. The facial recognition in surveillance and anomaly detection in financial transactions are examples of such integration that can be used in real-time [15]. Besides, the role of containerization and DevOps in the mechanism of making real-time data engineering a possibility cannot be overestimated. One can use Docker and Kubernetes technologies to make sure that the same environment of deployment is applied during the process of development, testing, and production. These tools will be used in combination with constant integration and constant deployment (CI/ CD) pipelines to ensure data pipelines and analytical models are tested and deployed in the shortest possible time without disrupting the uptime of the systems [16]. Security technologies are also important, particularly in real-time systems that have sensitive information such as health or finances. The norms include intrusion monitoring in real time, role-based policies and access control, and encrypting the rest and transit. Real-time data architecture is integrated with such technologies as TLS, OAuth, and SIEM (Security Information and Event Management) systems to maintain the confidentiality, integrity, and availability [17]. Since the technological pioneers of real-time data engineering have been discussed, there is a need to possess illuminating knowledge of the real-life implementation of these systems in various smart applications. This comes in handy to create the context of the discussion for the real-world case and illustrate the practical worth and the outcomes of correctly designed data pipelines.

4. Applications of Real-Time Data Engineering in Smart Systems

Having mentioned the technical and architectural foundations of real-time data engineering, it is time to analyze how such systems are applied to practice in smart surroundings. The implementation of smart systems nowadays is based on real-time data pipes and is proactively and intelligently reacted to in a broad spectrum of applications, such as healthcare, transportation, manufacturing, finance, and urban infrastructure. Smart patient monitoring systems with real-time data engineering in healthcare can calculate vital signs data from wearable devices and sensors in hospitals to detect an abnormality, such as a cardiac arrest or respiratory failure. These systems

take real-time information and utilize predictive models to forecast adverse outcomes, which may prove helpful to intervene as soon as possible and reduce mortality rates [18]. They are also real-time, and this may be extended to epidemic surveillance where real-time data streams across multiple hospitals, laboratories, and any other open-ended sources can be aggregated to identify and act on disease outbreaks almost in real-time [19]. Smart transport is another field that has been transformed with the real-time capability of data. Intelligent Transport Systems (ITS) refer to systems that utilize GPS sensors, traffic sensors, and vehicle telemetry to detect the conditions of the road and the movement of the traffic to facilitate the control of traffic and autonomous movement of vehicles. Real-time data streams are also handled to detect congestion and dynamically regulate traffic lights and recommend other paths to traffic. Furthermore, autonomous vehicles are premised on information consumption and processing of LiDAR, radar, and cameras that make decisions in seconds, i.e., braking or switching lanes, in response to environmental stimuli [20]. Industry 4.0 has been incorporated at the industrial level, with real-time data engineering introduced to smart manufacturing. Production lines are equipped with sensors of IoT sensors and are used to collect information on temperature, vibration, humidity, and the usage of machines, which are analyzed in edge and cloud analytics. Such systems expect equipment breakdown and pre-plan the maintenance and optimize production schedules to minimize lost time and maximize efficiency. Even a few seconds' delay in the processing of such cases can lead to defects in the products or system malfunction, and hence the requirement to reach real-time is essential [21]. The financial services have been at the forefront as far as the use of real-time information solutions is concerned. The high-frequency trading systems have to have the capability to manipulate market data, news feeds, and other economic indicators within less than a microsecond to execute a trade before their rivals. Similarly, the fraud detection systems scan the flow of transactions as they are run to identify any suspicious patterns, such that unauthorized transactions may be blocked before processing them. The success of such applications depends on the seamless integration of streaming analytics, machine learning models, and alert systems with good data engineering practices [12]. The other domain where real-time data systems are being implemented is in urban planning and smart cities. The city management platforms merge the data of surveillance cameras, air quality sensors, weather sensors, and citizen reports to provide the city authorities with a dynamic and integrated view of the city. The traffic and level of pollution, and the state of the infrastructure, are also provided in real-time on dashboards, in which an individual can decide more quickly and in a more knowledgeable manner. One such technology is the smart street lights that get brighter when there are people nearby, and this makes the streetlights safer, besides saving energy [3]. Smart retail is also assisted by real-time data engineering. In order to tailor the offers and manage the stock levels, retailers equip the systems that track customer movements, purchase transactions, and stock levels. These systems must be capable of processing information in real time in order to give an offer at the right time or to re-order goods before they go out of stock. The more AI is applied, the more accurate and relevant the recommendation engines can be when discussing the products they suggest to their customers based on real-time customer data, which ensures a greater number of interactions with customers and satisfaction [14]. Smart farming is gaining a lot of significance as an application in the agricultural industry. Farmers can schedule their irrigation, fertilization, and harvest by being informed in real time regarding the soil sensor, drones, and weather information. It is possible to make correct interventions to raise the crop yield and reduce resource use with the help of machine learning algorithms run on such data. These systems can improve productivity besides support environmental sustainability by cutting down on the wastage. Despite the examples presented of the wide-scale impacts of real-time data engineering, there has been the articulation of advanced problems in terms of scale, latency, data quality, and system durability. The following section discusses these limitations and their potential solution in a real-life deployment Shown in Figure 2 Applications of Real-Time Data Engineering in Smart Systems.

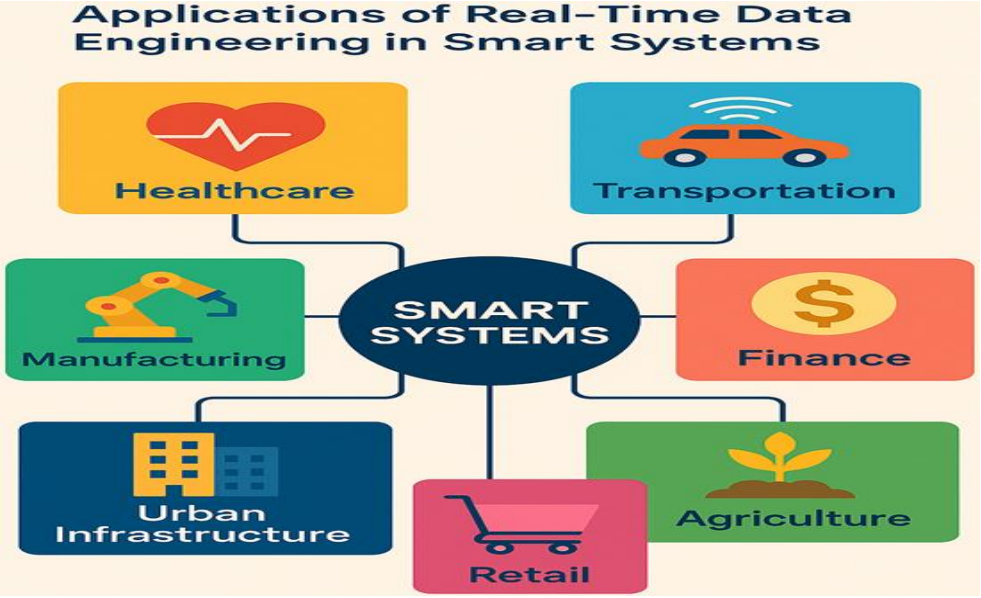

Figure 2 Applications of Real-Time Data Engineering in Smart Systems

5. Challenges and Limitations in Real-Time Data Engineering

As much as it is factual that real-time data engineering may be a revolutionary idea, there are numerous technical, operational, and strategic problems with the deployment of such systems on a large scale. These issues cut across data quality, system latency, scalability, fault tolerance, security, and balancing real-time and batch processes. One of the most important challenges is the assurance of real-time quality of data. Streaming data is noisy or incomplete, or intermittent, particularly in cases where it is acquired with the help of heterogeneous devices that operate in dynamic circumstances. The traditional data cleaning and transformation methods that are generated to run in batches do not work well in real time. It therefore means that organizations must consider adopting lightweight and high-throughput preprocessing systems, which is a trade-off between speed and accuracy of the data [16]. The guarantee of real-time processing would be based on the ability to reach sub-second response times, which is hard to accomplish when processing large and complex data. The reason for the latency increase can be network delays, serialization overheads, back pressure at message queues, etc, all of which are undesirable to smart systems. They can be solved with the help of such methods as windowed computation, data partitioning, and load balancing; however, they should be well-tuned and continually monitored [7, 17]. Latency is not only closely associated with scalability, but it is also challenging to use in such a situation, where the speed of data is rather unpredictable. We have a probability of an explosion in data volumes in smart cities or in the viral application of social media. The systems must be scalable horizontally, necessitating a stateless design, container orchestration, and auto scaling infrastructure that dynamically varies the resource allocation based on workload trends. Live pipelines cannot support fault tolerance, and it is an obligatory feature in mission-critical systems like healthcare and finance. Any failure in gathering, disseminating, or processing information would lead to missed events, ineffective outputs, or halting of the system. The distributed architecture that is required is resilient on the basis of inherent redundancy, failover, and exactly-once semantics of processing. This also makes it hard to design the system and involves a properly designed testing and monitoring system [9-12]. Security and privacy are other challenges. Real-time systems tend to process sensitive data, which should not undergo unauthorized access, interception, or tampering. The reduction of latency must not be made during the implementation of the real-time encryption and access control policy, as well as secure data transmission protocols. Also, the privacy regulations (GDPR and HIPAA) are very

stringent in the way they are managed, and one must integrate compliance mechanisms into the streaming procedures. Maintenance and development of real-time pipelines is also a challenge, which is brought up during operation. Streaming pipelines cannot be halted and revised, as is the case with a batch system. The process of updating the logic process, analysis models, or altering the schema must be without disrupting the data flow. This leads to the use of sophisticated deployment strategies such as blue-green deployments, feature toggles, and schema evolution by backward compatibility. It is a strategic issue that actually requires processing in real time in a system. All this does not necessarily require instant processing, and blind application of real-time architectures can lead to unnecessary complexity and cost. The organizations must decide on the business value of immediacy and must develop mixed systems that combine real-time and near-real-time and batch processing where appropriate. These limitations imply that the field of data engineering needs constant innovativeness and growth. In the next section, we will comment on the emerging trends and research directions aimed at overcoming those problems and broadening the scope of possibilities of real-time smart systems.

6. Future Trends in Real-Time Data Engineering for Smart Applications

Due to the new growth of organizations that create smart systems, the field of real-time data engineering is being influenced by a variety of new tendencies and research discoveries. The trends are not just the reactionary measures to the challenges mentioned above, but rather proactive trends of the creation of smarter, self-reliant, and context-related data systems. Integrating machine learning, federated architecture, data mesh paradigms, and quantum computing will transform reality on what is possible in real-time pipelines of data. One of the most important relocations is the one related to the integration of real-time machine learning and automated management of models into data pipelines. As the role of dynamism in settings grows in importance, structures of deployment of models, as with those of a fixed and trained set of models, are being replaced by systems that are able to learn on the fly and learn with time. This means such systems can be faithfully updated to the current data, with new information as it is available, and thus can still be accurate, even in concept drift situations with the trends of data over time. Such development requires real-time feature engineering, automatic data validation, and monitoring of the data pipeline models per se [1]. The future of federated data architecture is also bright. The latency is also eliminated in federated models since the processing of the data may be carried out on the edge devices or on an organizational level, unlike centralized systems, and the problem of data privacy can be solved. Real-time federated learning model allows a group of devices or other data silos to learn models together, without exchanging actual data, which is suitable in the setting of healthcare, finance, and IoT ecosystems. These forms of distribution systems require a high degree of coordination and effective communication systems to coordinate updates and consistency in the system [2, 3]. The other groundbreaking paradigm is the data mesh, which suggests a decentralized type of data architecture that internalizes the data as a product and allows cross-functional teams to possess and manage their data pipe. Domain teams create live pipelines that are used and operated, in contrast to a central data engineering organization that encourages scalability and agility. It shares a similar methodology with DevOps and microservices and needs to be reliant on a broad range of standardized APIs, observability, and self-service platforms to function. Real-time data engineering is also being affected by the proliferation of low-code and no-code platforms. The platforms can allow business users and analysts to build data streams, dashboards, and simplistic real-time models without much knowledge of the program. Although they are not applicable in a very complex use case, they also reduce the barrier to real-time analytics entry in a small enterprise business or a department considerably. The guidelines with artificial intelligence are being added to the drag-and-drop tools that are simplifying the pipeline building and maintenance process as well [14]. Quantum computing is a highly emerging field that is being considered to enable real-time data engineering in the future. The amount of processing that the quantum systems can perform is exponential and would be capable of converting stream processing and, more specifically, complex optimization and real-time simulations, which is

computationally infeasible currently. The general-purpose quantum processors will not come into being for years later but the hybrid quantum-classical schemes are already being tested on real-time logistics optimization and fraud detection. Streamless processing of serverless is still a developing infrastructure, and is likely to reinvent the economy and scale of real-time pipes. The serverless model fully gets rid of the infrastructure management and allows developers to concentrate on the business code, and be automatically scaled up and down according to demand. Innovation to solve the cold-start latency and more advanced semantics of the processing process is also being made in this region to make it a more generalized usage in smart applications [6-9]. There is also a focus on the ethical issues of AI and their explicability. Real-time interpretation and justification of decisions is obligatory because autonomous systems are capable of applying real-time decisions in their areas of work, including law enforcement, healthcare, and finance. The next generation pipelines would also require them to contain real-time explainability systems that would enable the user to have a clear understanding of how the models function and their confidence and provenance of data [7]. Finally, multi-modular real-time pipeline perforation data fusion also exists. One stream can be used to combine audio, video, text, and sensor data, and, in this way, smart systems have a more contextual understanding of their environments. It is especially relevant to autonomous systems, smart surveillance, or immersive systems such as AR/VR that require making decisions on the basis of a huge number of different and simultaneous types of information. Such systems have an area of current research and development, which is the synchronization, bandwidth, and processing efficiency. These new trends point out that real-time data engineering will be smarter, autonomous, decentralized, and user-friendly in the future. The final part of this article provides the conclusion of the main findings and speculates on the long-term outcomes of the same industry and the society concerned by assuming that innovation is the main theme of this development.

Conclusion

One of the pillars of smart application is real-time data engineering, which enables a system to sense, assess, and respond to the world on a new level never experienced before. The position of real-time data processing has significantly expanded in scope and significance since its inception as an architectural tool to the innovative application in a range of fields, such as healthcare, transportation, manufacturing, and city planning. The paper has taken the chronology of real-time data engineering, starting with the simplest architecture, which is a combination of the ingestion, stream processing, storage, analytics, and delivery layers. One of the enabling technologies, which the article has already mentioned, for AI and containerization is to allow these systems to scale with a small latency. The multiple applications proved the practical impact of the skills in which lives are saved, cities optimized, and businesses transformed through the intelligent application of streaming data. This potential has however great challenges. The quality of data is not very easily maintained as well, latency is also low and must be maintained, scalability is not a trivial issue either and it must be addressed at a higher level, fault tolerance is also not a trivial issue and must be approached at an advanced level and finally, security is also not a trivial issue and must be addressed at an advanced level. The limitations highlighted in this case have illustrated the importance of appropriate planning, continuous observation, and experimentation in the development of real-time systems. In the future, real-time data engineering will be affected by a synergy of disruptive trends online learning, federated models, data mesh structures, quantum computing, and ethical AI, among others. These developments are able to introduce even smarter applications and make them more versatile, decentralized, and available, but also pose new requirements of transparency, fairness, and accountability. Eventually, as the idea of smart applications is incorporated into society, the work of data engineers, system architects, and policymakers also grows correspondingly. The present requirement is no longer to develop fast and scalable systems, but ethical, explainable, and resilient. The data engineering, as such, however, is not only a technical vocation, but a social need, and it determines the relations of man to machines and to other people surrounding him.

References

[1]. Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. Mobile networks and applications, 19(2), 171-209.

[2]. Duan, L., & Da Xu, L. (2024). Data analytics in industry 4.0: A survey. Information Systems Frontiers, 26(6), 2287-2303.

[3]. Hassan, A. A., & Hassan, T. M. (2022). Real-time big data analytics for data stream challenges: an overview. European Journal of Information Technologies and Computer Science, 2(4), 1-6.

[4]. Batty, M., Axhausen, K. W., Giannotti, F., Pozdnoukhov, A., Bazzani, A., Wachowicz, M., ... & Portugali, Y. (2012). Smart cities of the future. The European Physical Journal Special Topics, 214(1), 481-518.

[5]. Zaharia, M., Das, T., Li, H., Hunter, T., Shenker, S., & Stoica, I. (2013, November). Discretized streams: Fault-tolerant streaming computation at scale. In Proceedings of the twenty-fourth ACM symposium on operating systems principles (pp. 423-438).

[6]. Stojkoska, B. L. R., & Trivodaliev, K. V. (2017). A review of Internet of Things for smart home: Challenges and solutions. Journal of Cleaner Production, 140, 1454-1464.

[7]. Alam, M. A., Nabil, A. R., Mintoo, A. A., & Islam, A. (2024). Real-time analytics in streaming big data: techniques and applications. Journal of Science and Engineering Research, 1(01), 104-122.

[8]. García-Gil, D., Ramírez-Gallego, S., García, S., & Herrera, F. (2017). A comparison of scalability for batch big data processing on Apache Spark and Apache Flink. Big Data Analytics, 2(1), 1.

[9]. Ziehn, A. (2020). Complex event processing for the Internet of Things. fog, 1(3), 4.

[10]. Behera, R. K., Das, S., Rath, S. K., Misra, S., & Damasevicius, R. (2020). Comparative Study of Real-Time Machine Learning Models for Stock Prediction through Streaming Data. J. Univers. Comput. Sci., 26(9), 1128-1147.

[11]. Naayini, P. (2025). Building AI-driven cloud-native applications with Kubernetes and containerization. International Journal of Scientific Advances (IJSCIA), 6(2), 328-340.

[12]. Hartmann, M., Hashmi, U. S., & Imran, A. (2022). Edge computing in smart health care systems: Review, challenges, and research directions. Transactions on Emerging Telecommunications Technologies, 33(3), e3710.

[13]. Satyanarayanan, M. (2017). The emergence of edge computing. Computer, 50(1), 30-39.

[14]. Nastic, S., Rausch, T., Scekic, O., Dustdar, S., Gusev, M., Koteska, B., ... & Prodan, R. (2017). A serverless real-time data analytics platform for edge computing. IEEE Internet Computing, 21(4), 64-71.

[15]. Harrington, K. (2023). Real-Time Fraud Detection Using Machine Learning and Stream Processing.

[16]. Kuruba, M., Shenava, P., & James, J. (2018). Real-time DevOps analytics in practice. In Proc. QuASoQ (p. 40).

[17]. Razzaq, M. A., Gill, S. H., Qureshi, M. A., & Ullah, S. (2017). Security issues in the Internet of Things (IoT): A comprehensive study. International Journal of Advanced Computer Science and Applications, 8(6), 383.

[18]. Kang, K. D. (2022). A review of efficient real-time decision making in the Internet of Things. Technologies, 10(1), 12.

[19]. Fallatah, D. I., & Adekola, H. A. (2024). Digital epidemiology: harnessing big data for early detection and monitoring of viral outbreaks. Infection prevention in practice, 6(3), 100382.

[20]. Saaristola, T. (2022). Data collection system for autonomous vehicles.

[21]. Phuyal, S., Bista, D., & Bista, R. (2020). Challenges, opportunities, and future directions of smart manufacturing: a state-of-the-art review. Sustainable Futures, 2, 100023.